nuxeo / Content Management Platform For Business Applications

Nuxeo Enterprise Platform 5.3

Technical Documentation

Released on 02/11/13

1. Nuxeo EP 5.3 Documentation Center 2
1.1 Installation and Administration Guide 2
1.1.1 Nuxeo EP 5.3.2 Installation Guide 3
1.1.1.1 Hardware, Software and OS Requirements 3
1.1.1.2 Installing Nuxeo EP 7
1.1.1.3 Configuring Nuxeo EP 8
1.1.1.3.1 Available Parameters for nuxeo.conf (5.3.2) 12
1.1.2 Managing your application with Nuxeo RMI Shell (command line tool) 15
1.2 Set up the Nuxeo Admin Center 26

Nuxeo EP 5.3 Documentation Center

Welcome on Nuxeo EP 5.3 Documentation Center

This doc center refers to the 5.3 version of Nuxeo EP and explains how to install this version of Nuxeo EP.
You may want to check the documentation for the latest version of Nuxeo EP for more information about the platform.

Download

Download this documentation in PDF.

License

This documentation is copyrighted by Nuxeo and published under the Creative Common BY-SA license. More details on the Nuxeo documentation license page.

Recently updated

- Set up the Nuxeo Admin Center
  updated Dec 07, 2012 (view change)
- Nuxeo EP 5.3 Documentation Center
  updated Dec 05, 2012 (view change)
- Managing your application with Nuxeo RMI Shell (command line tool)
  updated Sep 05, 2012 (view change)
- Installing Nuxeo EP
  updated Sep 05, 2012 (view change)
- Nuxeo Shell
  updated Dec 28, 2011 (view change)
- Available Nuxeo Distributions
  updated Dec 15, 2011 (view change)
- Available Parameters for nuxeo.conf
  updated Nov 16, 2011 (view change)
- About the content repository
  updated Sep 13, 2011 (view change)
- Platform features quick overview
  updated Sep 13, 2011 (view change)
- Repository access
  updated Sep 02, 2011 (view change)

Installation and Administration Guide

Table of Contents:

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

- Nuxeo EP 5.3.2 Installation Guide
  - Hardware, Software and OS Requirements
  - Installing Nuxeo EP
  - Configuring Nuxeo EP
- Managing your application with Nuxeo RMI Shell (command line tool)

Nuxeo EP 5.3.2 Installation Guide

Welcome to the Nuxeo EP Installation Guide.

Nuxeo EP is the base platform for Nuxeo products, here are common instructions for all products.

Nuxeo EP can be installed on any computer, providing that Java is installed.

Depending on the product, we provide various packagings: ZIP archive, Windows installer, Debian package, ...
The ZIP archive is the reference, other packages will install the same content as the ZIP archive adapting the deployment to the target platform.

We also provide those products embedded in various J2EE servers: JBoss, Tomcat, Jetty (for development only), ...
Even if the structure changes, the Nuxeo content is the same, install and configuration procedures are unchanged.

Detailed install instructions
• Hardware, Software and OS Requirements • Installing Nuxeo EP • Configuring Nuxeo EP   • Available Parameters for nuxeo.conf (5.3.2)

Join the Nuxeo community
• Use our discussion forums to share your experience, connect to other users and get help • Subscribe to the ECM list to stay informed about the evolution of the platform • Follow us on LinkedIn, Facebook and Twitter.

Hardware, Software and OS Requirements

This section presents information about the running environment for a Nuxeo server. Listing all required software, giving a recommended configuration and listing some others, known as operational, this sections aims at helping you to validate or define your production environment but the list is not exhaustive and needs to be completed with the users' experience.

Outline of this document:

- Requirements
  - Check your Java version
  - Install Java
    - For Windows users
    - For Linux users

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

- For Mac OS users
- Recommendations
  - Hardware configuration
  - Default configuration
  - For optimal performances
  - Known working configurations
    - OS
    - JVM
    - Storage backends
    - LDAP

Requirements

Nuxeo EP can run on Linux, Mac OS X and Windows operating systems.

All you need for a quick start is:

- Java Development Kit (JDK) (a JRE is not enough)

Nuxeo EP is fully based on Java. This means that Nuxeo can easily be deployed on any platform with a Java Software Development Kit (SDK).

It is required to use either Java 1.5 (also called Java 5) or Java 1.6 (also called Java 6). Java 6 is the preferred version, if you have the choice.

> ⚠ We recommend Sun's JRE and don't support other JDK.
> Don't hesitate to contact us if you need us to support a JDK from another vendor.

- Java EE server with EJB 3 support enabled, or simple servlet container.
  Nuxeo packages include either JBoss, Tomcat or Jetty.

Check your Java version

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

To check that you have the right version of Java

1. Open a terminal :
   - on Linux or Mac OS: open a terminal.
   - on Windows: press "windows" key + r, type "cmd" (or "command') in the run window and press "OK" or open the "Prompt command" from "Start > Programs > Accessories" menu.
2. Type "java -version" and press "Enter".
   If Java is correctly installed on your computer, the name and version of your Java virtual machine is displayed:

```
$ java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02-279-10M3065)
Java HotSpot(TM) 64-Bit Server VM (build 16.3-b01-279, mixed mode)
```

   If Java is not installed on you computer, it fails to display the java version. Then, you need to install Java.
   If Java is installed but not included in the PATH, it fails to find the java command. Then, you need to add $JAVA_HOME/bin/ in your PATH (see How do I set or change the PATH system variable?)

Install Java

For Windows users

If the required version of Java is not installed on your computer:

1. Download it from Sun website and choose the appropriate platform for your hardware and Windows version.
2. Run the downloaded .exe file and follow instructions.

For Linux users

On a modern Linux distribution, you can now install Sun Java 6 with the packaging system of your distribution (using rpm, apt, yum, etc.).

If the JDK from Sun is not available for your OS, you can still download it from Sun website.

⚠ Since Ubuntu 10.4 (Lucid Lynx), sun-java-6 is available from partners software sources (apt repositories).

For Mac OS users

An appropriate version of Java should already be installed on your computer if you are using either Mac OS X Leopard (Java 5) or Snow Leopard (Java 6).

Recommendations

Hardware configuration

Nuxeo is designed to be scalable and thus to be deployed on many servers. It can be installed on only one server for a start, and can also easily be installed on many servers. The constant is that there is the need to have one

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

modern server with good performances. Then the other servers can be more lower-end.

So the numbers below are given for the one needed high-end server.

- RAM: 2Gb is the minimum requirement for using Nuxeo
- CPU: Intel Core2 or equivalent and better
  You might be better avoid machines from the Intel Pentium 4 Xeon series since some models have a too small amount of cache. This impairs performance greatly compared to other CPU architecture of the same generation. Intel Pentium 4 servers are quite widespread because of an attractive price politic.
- Storage (disk) space: the minimum Nuxeo installation, along with the needed JBoss and libs, takes something between 200Mb and 250Mb on a filesystem. Then the final size will of course depend on the amount of data that will be stored in Nuxeo. A safe bet (until we provide better numbers) is to consider data space ratio of 1.5Gb to 2Gb.

Default configuration

The default persistence configuration is lightweight and easy to use, but it is not made for performance.

Nuxeo 5.3 uses:

- H2 for SQL Data (directories, JBPM, Relations ...)
- FileSystem persistence with VCS for Document repository

For optimal performances

- Linux 64 bits.
- PostgreSQL 8.4
  Use PostgreSQL for document repository and all other services.
  With Nuxeo 5.1/5.2, configure the document repository to externalize the blobs to filesystem.

Known working configurations

OS

- Debian GNU/Linux 5.0 Lenny
- Linux Ubuntu 32 and 64 bits: Edgy Eft (6.10), Feisty Fawn (7.04), Gutsy Gibon (7.10), Hardy Heron (8.04 LTS), Intrepid Ibex (8.10), Jaunty Jackalope (9.04), Karmic Koala (9.10), Lucid Lynx (10.04 LTS).
- Linux Mandriva 2008.1
- Unix
- Mac OS X Tiger (10.4), Leopard (10.5), Snow Leopard (10.6)
- Ms Windows 2000, Windows 2003 server 32 and 64 bits, Windows XP, Windows 7

JVM

- Sun JDK 1.5
- Sun JDK 6 (recommended)

Storage backends

Different backends may be set as well for Nuxeo Core repository as for all other Nuxeo services that persist data. Here is a list of known working backends:

- PostgreSQL 8.2, 8.3 and 8.4
- MySQL 5
- Oracle 10
- MsSQL 2005
- HSQL (this is default Nuxeo 5.1 embedded database).

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

- Derby (this is the default Nuxeo 5.2 embedded database).
- H2 (this is the default Nuxeo 5.3 embedded database).

LDAP

- OpenLDAP
- OpenDS
- MS Active Directory

Installing Nuxeo EP

Once you have checked that you have the required version of Java installed on your computer, you need to install Nuxeo EP.

Installing Nuxeo EP takes 3 steps:

1. install the downloaded package,
2. start the server,
3. connect to the application in a browser.

Install Nuxeo EP ZIP package

Just decompress it (double click on the file or run unzip from the command line).

Start Nuxeo EP

Nuxeo EP is distributed with a JBoss. There are startup executables for each OS.

Linux / Mac OS / Solaris

To start Nuxeo EP on Linux, Mac OS or Solaris:

1. Launch a terminal and go to your installation directory.
2. Make the scripts executable (some ZIP tools loose the executable bit):

```
chmod +x *.sh *ctl
```

3. Start the server using nuxeoctl script:

```
./nuxeoctl start
```

On Mac OS X, you can also use convenient commands from the Finder:

1. Make the .command scripts executable:

```
chmod +x *.command
```

2. Click from the Finder on "Start Nuxeo.command" or "Stop Nuxeo.command"

Windows

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

To start Nuxeo EP on Windows:

1. Open the file explorer and go to your installation directory.
2. Click on "NuxeoCtl.exe" then use the buttons in this Nuxeo control panel to start/stop the server and show/hide the logs.

If you installed Nuxeo as a Windows service (automatically done when using the Nuxeo installer for Windows), you can use the service for starting the server instead of the Nuxeo control panel.

Connect to the server using your web browser

1. Open a browser and type the URL `http://localhost:8080/nuxeo/`.
2. Log in to Nuxeo EP using the **Administrator** user name and the **Administrator** password.
   Once logged with superuser credentials, you can start creating regular users and groups and organize your folders.

The dedicated user guides will help you figure out the main features of Nuxeo products.

Configuring Nuxeo EP

Simple configuration with nuxeo.conf

Nuxeo is mainly configured by editing the nuxeo.conf file, located in the bin folder of your Nuxeo server.
Cutomizing the default nuxeo configuration takes 3 steps:

1. Copy the nuxeo.conf file.

   It is better, but not mandatory, to store your customized nuxeo.conf outside Nuxeo. This way, you will be able to easily upgrade Nuxeo, keeping your configuration safe apart of Nuxeo directory.

2. Edit it.
3. Start your server and tell Nuxeo where to find the customized nuxeo.conf.

Below are the instructions for Linux, Mac OS X and Solaris users, and for Windows users.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

For Linux/Mac OS X/Solaris users

1. Copy bin/nuxeo.conf in your home directory.

```
cp bin/nuxeo.conf ~/
```

2. Edit nuxeo.conf with a text editor and set your preferences (see Available Parameters for nuxeo.conf (5.3.2)).

```
gedit ~/nuxeo.conf
```

   Some recommended changes are listed below.
3. Run nuxeoctl giving the path to your nuxeo.conf.

```
NUXEO_CONF=~/nuxeo.conf ./bin/nuxeoctl start
```

For Windows users

1. Using the file explorer move bin/nuxeo.conf to your desktop.
2. Edit nuxeo.conf with a text editor and set your preferences (see Available Parameters for nuxeo.conf (5.3.2)).
   Some recommended changes are listed below.

   Do not use Office writers, neither Notepad.
   Wordpad is fine, Notepad++ and SciTE are good text editors, there are a lot of other text editors

3. Run NuxeoCtl.exe.
   If NuxeoCtl.exe doesn't find a nuxeo.conf in the bin directory, it will automatically look for it on the user desktop.

Recommended parameters changes

Recommended changes are:

```
nuxeo.bind.address=server_IP

nuxeo.notification.eMailSubjectPrefix=[Nuxeo]

mail.smtp.host=localhost
mail.smtp.auth=true
mail.smtp.username=anonymous
mail.smtp.password=password
mail.from=noreply@nuxeo.com
```

Or, if you use a database template (see below "Advanced configuration with nuxeo.conf and templates"):

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
nuxeo.db.host=localhost
nuxeo.db.name=nuxeo
nuxeo.db.user=user
nuxeo.db.password=password
```

Advanced configuration with nuxeo.conf and templates

The templates in the directory "templates" are used to generate configuration and datasource files at server startup when it is not already configured.

Usage

General instructions

Edit nuxeo.conf and choose the name of the template(s) you want.

For instance, to switch from default configuration (with Derby database) to recommended PostgreSQL, replace

```
nuxeo.templates=default
```

with

```
nuxeo.templates=default,postgresql
```

Default values are used for undefined properties in nuxeo.conf.
Take care that the following sample is defining an empty property:

```
nuxeo.sample.value=
```

Do not edit nuxeo.defaults files under the templates directory except, if needed, the one in custom template.

Technical overview

A server is considered as already configured when it has a config directory.
When the config directory doesn't exist, templates will be used to generate all configuration files (config and datasources).

The template files contain defined parameters such as ${sample.parameter}.

Values for parameters replacement are calculated this way:

- If nuxeo.conf does not define nuxeo.templates, then nuxeo.templates equals "default" (the deprecated parameter nuxeo.template is still read for backward compatibility).

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

- The ${nuxeo.templates} value is used for determining the chosen template(s).
- For each value "nuxeo.template" of ${nuxeo.templates} (comma separated values, relative to "templates/" directory or absolute path), the corresponding file templates/${nuxeo.template}/nuxeo.defaults is read for defining new default values and maybe including other templates which are recursively parsed.
- The file templates/nuxeo.defaults is read for default values not already defined.
- The file nuxeo.conf is read for custom values (overwriting default values).

Configuration files are then generated by this way:

- For each comma separated value of nuxeo.templates and nuxeo.template.includes (let say sample.template), files in templates ${sample.template} are copied using the previously calculated values for replacing parameters.
- Every included template will potentially overwrite its precedents.

Customization

"custom" template allows to add customization such as using multiple databases, configuring services, ...

- Add your own template files in "templates/custom" directory.
  You can use either existing or new parameters in these new template files.
- Edit nuxeo.conf, set your parameters' values and set nuxeo.templates=custom
  You can refer to custom templates directory with a relative path or to your own custom templates directory with an absolute path.
- Edit custom/nuxeo.defaults and set nuxeo.template.includes parameter to define the list of existing templates to include (comma separated values); your custom template will be used at last.
  nuxeo.defaults files from included templates are also read.

In case you need multiple customizations, create multiple directories and reference them in a dedicated nuxeo.conf for each server.

Available templates

default

Default Nuxeo configuration.

Designed for development or test purpose.

Repository backend: H2

Services backend: Derby

postgresql (recommended)

Recommended configuration for production, based on PostgreSQL.

Think to add PostgreSQL JDBC driver into server/default/lib/ directory.

Repository backend: PostgreSQL XA

Services backend: PostgreSQL XA

mysql

Repository backend: MySQL XA

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

Services backend: MySQL

oracle

Repository backend: Oracle XA

Services backend: Oracle

stateful and stateless (deprecated)

Configuration separating stateful and stateless services (see Nuxeo documentation for details) using default values for database.

custom

Sample custom template. It's quite empty, feel free to play with it in order to create your own template.

> ✅ Best way is to create your own template directory outside Nuxeo directory (as for nuxeo.conf) then reference it with its absolute path.

Available Parameters for nuxeo.conf (5.3.2)

Here is a list of available parameters for nuxeo.conf. This list may not be exhaustive but it will be often updated.

Those parameters can be either environment parameters used by Nuxeo runtime or template parameters used for values replacement in configuration files.

Parameter	Default value ("/" separates default values for various templates)	Description
JAVA_HOME	None. If undefined nuxeoctl script will try to discover it.	Path to Java home directory.
JAVA_OPTS	-Xms512m -Xmx1024m -XX:MaxPermSize=512m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Dfile.encoding=UTF-8	Optional values passed to the JVM. Nuxeo requires at least 1024 Mo in JVM heap size and 256Mo as maximum permanent size (512 recommended). Decreasing garbage collector frequency avoid having too much CPU usage (Sun Java specific options, recommended by JBoss).
NUXEO_HOME	Parent directory of nuxeoctl. If any of the following variables exists, it will be used: JBOSS_HOME, CATALINA_HOME, JETTY_HOME	Nuxeo home directory.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

nuxeo.log.dir	log	Log directory (relative to NUXEO_HOME).
nuxeo.pid.dir	bin	Directory where to store Nuxeo PID file.
nuxeo.data.dir	data	Nuxeo data directory.
nuxeo.force.generation	true	If true, will force generation of configuration files; otherwise they are only generated when not existing.
nuxeo.templates	default	Comma separated list of templates to include. Templates paths must be absolute or relative to $NUXEO_HOME/templates/
nuxeo.bind.address	0.0.0.0	(JBoss only) Server binding address. "0.0.0.0" means "all available network interfaces".
org.nuxeo.ecm.instance.name	Nuxeo 5.3	Server name.
org.nuxeo.ecm.instance.description	Nuxeo ECM server	Server description.
org.nuxeo.ecm.product.name	Nuxeo DM	
org.nuxeo.ecm.product.version	5.3.2	
org.nuxeo.ecm.webapp.dashboard.mode	auto	
templateName.target	server/default/deploy/nuxeo.ear	Directory where *templateName* files will be deployed.
mailservice.user	nobody	(JBoss only) User for mail authentication.
mailservice.password	password	(JBoss only) Password for mail authentication.
mail.store.protocol mail.transport.protocol	pop3 smtp	(JBoss only) Server protocol parameters for mailing.
mail.user	nobody	(JBoss only) User who will receive mail (unused in Nuxeo).

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

mail.pop3.host	pop3.nosuchhost.nosuchdomain.com	(JBoss only) Mail server.
mail.debug	false	(JBoss only) Enable debugging output from the javamail classes.
nuxeo.notification.eMailSubjectPrefix	[Nuxeo]	Subject prefix in Nuxeo notification mails.
mail.smtp.host	localhost	SMTP gateway server.
mail.smtp.port	25	Mail server port.
mail.smtp.auth	true	
mail.smtp.username	anonymous	
mail.smtp.password	password	
mail.from	noreply@nuxeo.com	The address mails will be sent from.
nuxeo.db.name	nuxeo / NUXEO	Database name.
nuxeo.db.user	sa / nuxeo	Database username.
nuxeo.db.password	/ password	Database password.
nuxeo.db.host	localhost	Database host URL.
nuxeo.db.port	3306 / 1521 / 5432	Database host port.
nuxeo.db.min-pool-size	5	Database minimum pool size for Nuxeo datasources.
nuxeo.db.max-pool-size	20	Database maximum pool size for Nuxeo datasources.
nuxeo.vcs.min-pool-size	0	Database minimum pool size for Nuxeo repository (VCS).
nuxeo.vcs.max-pool-size	20	Database maximum pool size for Nuxeo repository (VCS).
nuxeo.url	http://localhost:8080/nuxeo	Application URL.
org.nuxeo.ecm.contextPath	/nuxeo	(Tomcat only) Application context path

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

org.nuxeo.ecm.platform.transform.ooo.host.name	127.0.0.1	DEPRECATED.
org.nuxeo.ecm.platform.transform.ooo.host.port	8100	DEPRECATED.
org.nuxeo.ecm.platform.transform.ooo.version	2.2.1	DEPRECATED.
org.nuxeo.ecm.platform.transform.ooo.enableDaemon	true	DEPRECATED.
opensocial.gadgets.host	localhost	
opensocial.gadgets.port	8080	
opensocial.proxy.proxySet	false	
opensocial.proxy.proxyHost		
opensocial.proxy.proxyPort		
opensocial.proxy.user		
opensocial.proxy.password		

Managing your application with Nuxeo RMI Shell (command line tool)

Deprecation Warning

The Nuxeo RMI Shell is deprecated since the version **5.4.0.1** of Nuxeo. You should consider using the Nuxeo HTTP Shell instead.

Overview

The Nuxeo Shell is a command line tool for everyone who needs simple and quick remote access to a Nuxeo server. You can see it as the Swiss army knife for the Nuxeo EP world. It can be used by regular users to browse or modify content, by advanced users to administrate Nuxeo servers, or by programmers to test and debug.

The Nuxeo Shell is designed as an intuitive and extensible command line application. It can both serve as a user, administrator or programmer tool, or as a framework to develop new Nuxeo command line clients.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

The application is based on the Nuxeo Runtime framework and thus offers the same extension point mechanism you can find on the server application.

The main features of the command line tool are:

- Two operating modes: an interactive and a batch mode.
- Advanced command line editing like:
- auto-completion
- command history
- command line colors
- command line shortcuts like: CTRL+K, CTR+A, etc.
- JSR223 scripting integration. You can thus connect and execute commands on a Nuxeo server in pure scripting mode.
- Extensibility - using Nuxeo Runtime extension points
  The advanced command line handling is done using the JLine library.

The only requirement is Java. The tool works on any Linux-like OS, on Windows and on Mac OS X.

- Overview
- Download Nuxeo Shell
- Build Nuxeo Shell
- User Manual
  - Command Options
    - Global Options
  - Commands
    - interactive
    - help
    - log
    - connect
    - disconnect
    - ls
    - tree
    - cd
    - pwd
    - view
    - rm
    - mkdir
    - put
    - putx
    - get
    - getx
    - export
    - import
    - script
    - chperm
    - useradd
    - groupadd
    - groupmod
    - select
    - lstypes
    - viewtype
    - lsusers
    - lsgroups

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

        - viewuser
        - viewgroup
- Troubleshooting
    - Check listened IP
    - Check connected server
    - Multi-machine case
- Extending the shell
    - Registering New Custom Commands
    - Java Code for the new commands
    - Building the shell plugin
    - Deploying the shell plugin

Download Nuxeo Shell

Since Nuxeo 5.3, Nuxeo Shell is included in most of distributions.

You can download latest released version from here. Previous versions are available from this page.

Build Nuxeo Shell

Nuxeo Shell is built with Maven from http://hg.nuxeo.org/nuxeo/nuxeo-distribution/

```
mvn clean install -Pshell
```

Result is in nuxeo-distribution-shell/target/ as a ZIP file.

User Manual

The Nuxeo Shell application is packaged as a zip archive. To install it, you need to unzip and copy the content in a folder.

> See README.txt in Nuxeo Shell for detailed information and usage samples

You can run the application in two modes:

1. Batch mode - in this mode you specify the command that will be executed. After the command execution the process will exit. The command has to be passed as the first argument.
   Here is an example of a command executed in batch mode:

   ```
   $ ./nxclient.sh -console export /path/to/remote/doc /path/to/local/folder -h
   192.168.2.54 -u Administrator -P AdminPassword
   ```

   Here is an example of a command executed in batch mode while also passing parameters:

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
$ ./nxclient.sh -console ls --host 192.168.2.54 -u Administrator -P
AdminPassword
```

2. Interactive mode - in this mode you can share a single session to run multiple commands against the remote Nuxeo EP.
   To start the application in that mode you should run the **interactive** command in one of two ways:

```
$ ./nxclient.sh interactive
```

```
$ ./nxshell.sh
```

   After entering in interactive mode, a command prompt will be displayed. At this prompt you can enter commands in the same way you specify them on the command line in batch mode.
   When not connected to a server, the prompt will be displayed as: `|>`.
   After connecting to a server named, let's say "nuxeo-platform", the prompt will be displayed as: `|nuxeo-platform>`.
   So, as we've seen in the above examples, executing a command is done by using the following syntax: `command [options] [parameters]` where options and parameters are optional and are specific to the command you run.

Example:

```
import -u /path/to/doc /path/to/local_file
```

In this case `"import"` is the command, `"-u"` is a flag option, `"path/to/doc"` and `"/path/to/local_file"` are both parameters.

Parameters are stand alone arguments (they are not bound to a specific option) and can be retrieved programmatically using their index. In the example above, the first parameter will have the index 0 while the second the index 1.

Command Options

Command options are defined by a name and optionally a shortcut (a short name). When referring to an option using its name you should prefix it by two hyphens `'--'`. When using short names you should only use one hyphen as a prefix.

For example if you have an option named `"host"` and using a short name of `"h"` the following commands are equivalent:

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
$ ./nxshell.sh interactive -h localhost

$ ./nxshell.sh interactive --host localhost
```

Options may take values or may be used as flags turning on/off a specific option by their simple presence.

When using long names you should specify the option values immediately after the option. However when using short names you can group options together.

Let say for example we have a command that support four options: `a, v, i, o`. `a` and `v` are flag options and both `i` and `o` take an argument as value. In this case, you can group options like the two following examples:

```
command -avi in_file -o out_file

command -avio in_file out_file
```

.

You should keep in mind that when grouping options taking arguments, those arguments will be assigned to each of these options in the order they were specified on the command line.

Global Options

Besides the specific options that commands may define, there are several global options applying to all commands. They are:

- `host` (**--host** or **-h**) the Nuxeo EP host where to connect to - defaults to localhost
- `port` (**--port** or **-p**) the Nuxeo EP port where to connect to - defaults to 62474
- `username` (**--username** or **-u**) the username to use - defaults to the "system" user
- `password` (**--password** or **-P**) the password to use
- `debug` (**--debug** or **-d**) to start with debug mode (logs at DEBUG level)

Commands

Here is the list of all built-in commands of nuxeo shell.

Notes:

- At the time of this writing, some of these built-in commands are not yet implemented. Those are be marked with an asterisk (*).
- The commands can be grouped in two categories:
  - offline commands - not needing a connection to work. For instance: help, log etc.
  - online commands - requiring a connection to work. These commands are automatically connecting if no connection is currently established.
- Some commands for which offline/online make no sense and are not available in both modes (batch and interactive). This will be specified in such a case.

Look at the included README.txt for an up-to-date list of available commands.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

interactive

Runs in the interactive mode. This command is not available when already in interactive mode.

Has no specific options:

```
$ ./nxshell.sh interactive
```

help

Displays the help page.

Takes an optional parameter which is the name of a command.

By default, displays the global help page. If a command is given as an argument, displays the help page for that command:

```
$ ./nxshell.sh help ls
```

log

Manage logs from console without having to edit log4j.xml. Allow to switch on/off a debug mode, add/remove loggers, ...

`log filename [log level [package or class]]` creates a file (in `$PWD/log/` if **filename** does not contain path separator, else given path is used). It will record logs at level greater or equal to **`level`** (if specified, else default level is INFO; available values are TRACE, DEBUG, INFO, WARN, ERROR, FATAL). If a **package or a class** (canonical name) is specified, logs will be filtered from this package/class.

`log off [package or class]` stops logging. Applies on all custom appenders (console and filenames) if no **`package`** or **`class`** is specified. Don't worry about the warning **log off** command may cause (the logger is re-created): {{log4j:WARN No appenders could be found for logger (org.nuxeo.ecm.shell.commands.InteractiveCommand). log4j:WARN Please initialize the log4j system properly.**}}**

`log debug` switches debug mode on/off. Same result as starting the shell with **-d** (**--debug**) option. This decrease the log level to DEBUG on console and filenames.

See log4j documentation for more information.

Available only in *interactive* mode

connect

Connects to a given server. If a connection is already established, close it first.

Available only in *interactive* mode

disconnect

Disconnects from the current connected server. If no connection exists, does nothing.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

Available only in *interactive* mode

ls

Lists the children of the current folder in the repository.

By default, Folderish types are displayed in blue.

Note that the **ls directory-name** command is accepted but won't list the content of directory-name, it will only list the content of the current directory. This might be improved in the future to provide a behavior alike the one of the Unix ls command.

Available only in *interactive* mode

tree

Displays the complete tree structure of the documents as it would be returned by the Unix **tree** command.

Available only in *interactive* mode

cd

Changes the current directory in the repository (to a Folderish document)

pwd

Displays the current path in the repository.

view

Views info about document. The information displayed can be controlled using these command options:

--all (-a) - view all data

--system (-s) - view only the system data

--acp - view the ACLs on that document

rm

Removes a document or a tree of documents.

mkdir

Creates a Folder document.

put

Uploads a blob to a file document. If the target document doesn't exists, creates it.

putx

Creates a document other than a file. Metadata (and blobs) are specified in the Nuxeo export format.

get

Downloads the blob from a File document.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

getx

Gets a document as an XML file (as export, but only for a document).

export

Exports documents from the repository.

import

Imports documents into the repository.

script

Makes it possible to run external scripts. Here are examples of such scripts.

From interactive mode use:

```
script --file <path_to_your_script> <args>
```

chperm

Changes a privilege on the given document.

useradd

Creates new user(s).

To create user joe:useradd joeTo create the users define in the users.csv CSV file:

```
useradd --file users.csv
```

groupadd

Creates new group(s).

groupmod

Modifies a group

To add user joe to the company-service1 group:

```
groupmod --user joe company-service1
```

To set the users of the company-service1 group to be only joe:

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
groupmod --set --user joe company-service1
```

To add the users in the users_for_group.csv CSV file to the company-service1 group:

```
groupmod --file users_for_group.csv company-service1
```

To set the users of the company-service1 group to be the users define in the users_for_group.csv CSV file:

```
groupmod --file users_for_group.csv company-service1
```

select

Search the repository using the NXQL query language. For example:

```
select * from Document where ...
```

This command can be used as a starting point to write another script to perform search service queries if needed.

This command is only available in the Nuxeo 5.2 branch.

lstypes

viewtype

lsusers

lsgroups

viewuser

viewgroup

Troubleshooting

Check listened IP

Be sure on what precise IP address and what precise port the server is listening to. To find out, use the **netstat** com mand on the server host. The example below shows that, on the same host, there are 2 servers listening on the 62474 port but on different IP addresses. You should always use as arguments to the nuxeo shell the exact IP address and port you find out.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
$ sudo netstat -ntlap | grep 62474
tcp        0      0 ::ffff:192.0.0.11:62474     :::*                        LISTEN
3623/java
tcp        0      0 ::ffff:192.0.0.10:62474     :::*                        LISTEN
1346/java      Alternatively you can use the *lsof* command.$ sudo lsof \-i :62474
COMMAND   PID       USER   FD   TYPE    DEVICE SIZE/OFF NODE NAME
java     3623      jboss  266u  IPv6 0x5d69a70      0t0  TCP []:62474 (LISTEN)
java     1346      jboss  266u  IPv6 0x5d7af30      0t0  TCP []:62474 (LISTEN)
```

Check connected server

Be sure to be connected on the right server. To do so, issue a **view** command. This will display information on the current context:

```
192.0.0.10> view
------------------------------------------------------------------------
UID: 2a13db70-f133-473c-9a90-6838d01610aa
Path: /
Type: Root
------------------------------------------------------------------------
Title: 2a13db70-f133-473c-9a90-6838d01610aa
Author: null
Created: null
Last Modified: null
------------------------------------------------------------------------
Description: 2a13db70-f133-473c-9a90-6838d01610aa
------------------------------------------------------------------------
```

Multi-machine case

When working with multi-machine installation, connect the Nuxeo Shell on the server running Nuxeo Core (i.e. the stateful server on a bi-machine configuration).

Extending the shell

If you a need a new command (a new functionality) not provided by the Nuxeo Shell, you can add it very simply by writing a Nuxeo plugin in exactly the same manner you would for a Nuxeo server instance. This is done by writing a Java class for each new shell command and declaring (that is registering) each command in a single XML file as a contribution to an extension point.

Here is an example of how to register two imaginary new custom commands **`addapplicants`** and **`purgeobsoletedocs`**.

Registering New Custom Commands

This is how to register the new commands:

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

```
<?xml version="1.0"?>
<component name="com.company.nuxeo.shellcommands" version="1.0">

  <documentation>Extra Nuxeo Shell commands shown as examples for the Nuxeo
Book.</documentation>
  <extension target="org.nuxeo.ecm.shell.CommandLineService" point="commands">

    <command name="purgeobsoletedocs">
      <description>Purge obsolete documents considering based on predefined
hard-coded logic.</description>
      <help>Purge obsolete documents</help>
    </command>

    <command name="addapplicants">
      <description>Adds applicants by creating the right folders with the right
permissions.</description>
      <params>
        <param index="0" type="path" />
      </params>
      <help>Adds applicants</help>
    </command>

  </extension>

</component>
```

Java Code for the new commands

This is how to write a Java class for a new command:

```
public class ApplicantsAdderCommand extends AbstractCommand {

    public static final String COMMAND_NAME = "addApplicants";

    public void run(CommandLine cmdLine) throws Exception {

        String[] elements = cmdLine.getParameters();

        // parse command line

        if (elements.length != 1) {
            log.error("Usage: " + COMMAND_NAME + " file");
            return;
        }
        File file = new File(elements[0]);
        addApplicants([]);
    }

    void addApplicants([]) {
    }

}
```

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.

Building the shell plugin

Finally the pom.xml file for the plugin to the Nuxeo shell needs to at least contain the following dependency to be able to build extend the AbstractCommand class. Building the plugin will generate a XXX.jar file. Note that you should of course replace the *version* given as an example with the version suited for your need.

```
<dependencies>
 [...]
  <dependency>
    <groupId>org.nuxeo.ecm.platform</groupId>
    <artifactId>nuxeo-shell-commands-base</artifactId>
    <version>5.1.7-SNAPSHOT</version>
  </dependency>
 [...]
</dependencies>
```

Deploying the shell plugin

1. Install or decide on using an already installed Nuxeo Shell.
2. Copy the generated XXX.jar file into the bundles directory of the Nuxeo Shell.
3. Delete the Nuxeo Shell data directory, if there is any. This is for purging any cache of registered JARs. The next time the Nuxeo Shell will be started, your new commands will be available.

Set up the Nuxeo Admin Center

To set up the Nuxeo Admin Center for Nuxeo DM/EP 5.3.2, follow these instructions.

Since 5.4, Nuxeo Admin Centre is automatically included in Nuxeo product releases.

To set up the Admin Center:

1. ADMIN:Download the zip that contains the required plugins (5 files with the .jar extension).
2. Unzip all the jar files in the correct folder:
   - JBoss distribution: `$HOME_NUXEO/server/default/deploy/nuxeo.ear/plugins`
   - Tomcat distribution: `$NUXEO_HOME/nxserver/bundles`
3. Download the library ADMIN:nuxeo-connect-client and copy it to the lib folder:
   - JBoss distribution: `$HOME_NUXEO/server/default/deploy/nuxeo.ear/lib`
   - Tomcat distribution: `$NUXEO_HOME/nxserver/lib#`
4. Restart your server.
5. Connect with Administrator credentials and click on the Admin Center link, in the header.
6. Register your instance with the Nuxeo Connect portal.

Copyright © 2010-2013 Nuxeo.
This documentation is published under Creative Common BY-SA license. More details on the Nuxeo Documentation License page.